How to start a test automation framework and not die trying

Juan Delgado, Isaac Méndez

jddelgado@up.edu.mx, isaac.mendez@up.edu.mx

Abstract

Nowadays, some organizations still execute the test life cycle of their projects without test automation, which means that they also do it without the possibility of implementing the best Quality Assurance & Test Automation practices..

If you would like to start a test automation effort in your organization with a formal and mature process, this paper will help you to start and not die trying!

In the process, this paper will introduce and discuss the Android Open Source project "UP Automation Framework" a wrapper of libraries developed by the Panamerican University (Universidad Panamericana in Spanish), that will help you implement a mobile test automation framework and structure in your organization. This paper explains the fundamental principles and demonstrates how to apply them. We'll look at identifying what type of mistakes are most frequently made in your organization and choosing from a toolkit of prevention and detection practices to address these mistakes. We'll also talk about how to choose the practices that best fit your organization's software development process.

Biography

Juan de Dios Delgado has more than 15 years of experience as engineer. He has worked in all kind of IT activities including development and support, and 7 years of experience working on testing. He has a strong knowledge on the IT and business areas with a big inclination to learning and analysis of Business processes. He has experience in Functional Testing, and has also worked with Agile and Waterfall methodologies, with certification as Scrum Master and ITIL Foundations.

He has participated in several projects, in the last years. He has worked on site in the US and Europe; near shore from Mexico to the US, and offshore from Mexico and India to Europe; working strongly with Functional Testing focusing on Automated Testing and Business Analysis

He is currently working at Mobica, an award-winning software development and integration services company with a global customer base.

Also, he's a Professor at the Panamerican University in Mexico, in the faculty of Engineering. He's focused in Software Quality Assurance and involved in the development and launch of interdisciplinary programs for students.

The rest of the team were students at the Panamerican University in the last year of the major in Artificial Intelligence Engineering. This project was part of the subject named Agile and Automated Testing Engineering.

Introduction

Testing is very important for software development companies because they need to guarantee their clients that their products are multiplatform, multidevice and that they will work correctly and efficiently. It is very important to perform testing during the entire development cycle and is essential to do it starting in the first stage of the project because this will reduce cost and time when fixing the bugs.

According to Oracle, industry surveys indicate that 75% of all functional testing is still done manually.[1] That's why we decided to develop this framework in order to make automation easier and save money for companies by reducing functional testing times.

Mobica, who employs Juan de Dios Delgado, is a company that works in software development for mobile devices, so that software must work across different platforms. That means that, alongside the development, a lot of testing must be built in order to assure any user will have a good experience with the software and that the client requirements will be achieved.

The company know that the quality of the software is very important, so they put a lot of effort in their testing. That comes with the usage of many different tools to make sure this testing is done in the best possible way. Juan found, however, that having this widely set of tools can be a little messy and overwhelming.

He also noted there are a lot of common tasks that testers want to automate, and they write scripts for, like opening apps or making calls. That means that there's a lot of work that is redundant, and that resources being used on something already existing when they could be used on something else. Surely a tool that gathered all these tools and had a set of these common actions could be made.

This paper narrates how the goal of making automation easier for Android devices was approached, with a project that was developed under the Scrum methodology by Juan de Dios and his group of students in a university in Mexico. The first name of the project was UP Android Automation Framework where UP stands for Universidad Panamericana (Panamerican University). Later, we decided that the name of the project will be AMX Framework where AMX stands for Automation Mobile Experience.

Android Mobile Devices for Testing

When beginning with this project about mobile devices, we had to decide what kind of devices we would work with. When making that decision, we found out that working with iOS would be very problematic, since we worked with the equipment already owned by the students, that consisted entirely in computers working with Windows 10 and Android mobile devices, so we defined our scope with these operative systems.

It is important to acquire a wide array of devices for your mobile Quality Assurance lab. Naturally, the underlying platform plays a big role in device procurement, as it will probably need a host of smartphones and tablets from different companies.

A good cross-section of screen dimensions, processors, and manufacturers (for Android) helps to ensure your team can test apps and websites in a multi-device, multi-platform and multi OS-testing environment. Checking for overall performance on older devices is vital as well.

While working on the development of this project, we decided to incorporate a Smartphone Test Farm to assure future users can do this easily.

Android Testing Concept

In testing, every test case is formed by many steps that must be achieved in order to determine if the test was successful or if it failed. In our project, a step will correspond to a certain task that has to be executed in the android device.

We run the test cases in different devices with the goal of obtaining statistics with strong foundations. There can be exceptions where we cannot execute some steps in a specific device, in this situation we must create a different test case or add the exception in the same test case.

When some step failed, the framework will log the failure with the description and the device where it occurred. It will also log when a step was executed successfully.

In order to achieve an accurate result of the test cases, we must run them many times depending on the type of application and on the type of testing that is required.

Methodologies and Process in class

We executed this project using an agile methodology. We have sprints with a duration of one week, but they were adjusted sometimes to suit specific requirements. We choose this duration because we have class once a week so in this way the team can work during the week and show their progress in the meeting we had at the beginning of the class.

We had two technical leads because we divided our team in two teams. This made the work easier for the technical leads and we got more features done in a sprint.

The technical leads worked along with the developers in the framework and in the reviewing of the code. The developers worked on the framework and on the creation of the database to save the information of the mobile devices. The three mobile testers worked on the test automation plan and the test cases that we used to create the demo.

The scrum master was on Bellevue, WA, and the rest of the team was on Aguascalientes, Mexico. We choose to work this way because the scrum master couldn’t travel to Mexico, but we take advantage of the technology to make this work with 3 meetings per week via video call and a daily communication between the scrum master and the technical leads.

This choice is preferable when you have members of the team living in other countries and they can’t travel very often, but we recommend all the team members to be collocated in the same city so they can communicate better and work smoothly.

The entire UP (Universidad Panamericana) AMX scope is described in a product backlog and is documented in Jira. We use this platform to create our Scrum dashboard where we have the historical information of all the sprints we had during the development of this project. We also have the effectivity of every sprint and we can check the sprints where we have more issues and the tasks we created in order to solve these issues.

The sprint planning prioritizes the remaining incomplete items in the product backlog and selecting the items for implementation in the subsequent sprint. At the end of each sprint, the backlog items which have been implemented in that sprint will be reviewed during the demo. Due to the duration of this project, we established that we must have a functional demo at the end of every sprint. The developers were required to do a functional testing on every task they did in order to expedite the review process. Then the demo was assembled by the technical leads which consist on a script that executed all the functions that were developed during the week and this was presented to the project manager.

Using an agile methodology helped us a lot during this project because we could adjust at any time depending on the issues or the requirements. Having a functional demo every week helped us to see our strong and weak spots in the planning and the development and we measured the scope of the project and how we could extended or remove tasks from it. The meetings helped us to review our progress, helped another members that had issues with their tasks and confirmed that we were achieving the requirements.

The members of the team that worked on this project are:

- Juan Carlos García and Ricardo Macías – Product Owners
- Juan de Dios Delgado – Scrum Master and Project Manager
- Isaac Méndez – Technical Lead
- Marco Montoya – Technical Lead
- Rodrigo Quiroz – Python Developer
- Ricardo Bustos – Python Developer
- Carolina Delgadillo – Python Developer
- Diego Camacho – Python Developer
- Luis García – Mobile Tester
- Jorge Ramírez – Mobile Tester
- Miguel Salazar - Mobile Tester

1.1 The structure of the team

The students worked in a conventional team used in a scrum project; where members with certain skillset can take specific tasks based on their experience or in the skills that they would like to develop.

Rather than having a specific set of students assigned to specific tasks, the team had a “groupthink” approach which allowed them to have a greater diversity and flexibility that gave them a wider experience and knowledge base in order to complete a task more quickly.

The challenge was to provide additional management over the students to verify the task was completed on the time to avoid any delay in the plan. The table below depicts the roles of our team.

Roles	Responsibilities
Product Owner	Responsible for the overall direction of the project
Scrum master (PM)	Responsible of ensuring a good relationship between the team and product owner and progress reporting
Consultant	Responsible for design, architecture and implementation of the AMX (Automation Mobile Experience) infrastructure.
SDET (Software Development Engineer in Test)	Responsible for the development of the core of the AMX Framework
Database Engineer	Responsible for the design and maintenance of the database
UI/UX Engineer	Responsible for the overall look & feel of the AMX
Tester	Responsible for testing the various portions of the application including end-to-end deployment

Tools

We used Jira to create and track our sprints. It was very useful because we could comment on the tasks to give an update of the progress or to give feedback. Also, this helped us to have a clear view of our goals and to estimate the number of sprints that we would need to finish all the tasks.

We used GitHub to manage our source code and to have a track of every change we made to the framework. We also used pull requests to review the code, which helped us to have a functional project in the repository with excellent quality code.

We used Zephyr, a JIRA add-on, to create our test cycles, this helped us to do a test cycle for every type of device and to detect bugs on a specific device or on a specific function.

We developed our framework using Python and the following libraries: uiautomator, pytest, validators, colorama, requests, firebase-admin, pylint, invoke, git-lint, pyyaml, pycodestyle, flask and flask_cors.

The demo was developed in Angular 7 and we used Firebase as our database motor.

UP (Universidad Panamericana) Automation Framework

1.2 History

This project was originally initiated in the second semester of 2018 by professors of the Panamerican University campus Aguascalientes in Mexico in response to a request from the engineering school to implement testing in mobile devices in the Software Quality Assurance class for students of Artificial Intelligence engineering.

The students had a gap as they did not previously have classes related to the quality of software in areas such as functional testing, automation testing, performance testing, API testing, etc. Also, there were not any classes related to agile methodology to work as a team in order to accomplish a project like this in real life so with this project they have gain the experience needed in their work.

The UP Framework was proposed and designed to provide the ability to gain experience in Manual and Automated testing using tools as Jira, Confluence, Python, STF.

The UP Framework can connect devices, create automated test cases and execute the test cases in the devices with a great user experience.

This execution on mobile devices can be used for a variety of reasons including but not limited to: user experience testing on a device, market analysis, performance, compatibility, application unit testing and so forth.

1.3 Business Case

The UP Framework was a unique college opportunity for the students in the following ways: it fulfills an immediate market need and it provides Panamerican University with its first intellectual property product as an Open source project. This allows the students to develop their skills in coding, scrum, communication, and team effort.

This framework can be offered to software companies to improve their testing process and for companies that does not have a testing department and they want to implement one. It is a software framework that can adapt to the needs of the client and it is very friendly with the user.

1.4 Implementation

This project was developed with Python and the code follows the style recommended by PEP 8. PEP 8 is Python's style guide with a set of rules for how to format your Python code to maximize its readability.

We created a diagram of the workflow of the framework and based on this we created the classes and the methods that we needed in order to get a functional first version.

We choose a non-relational database hosted on Firebase because this gives us the possibility to save all the data from the device and from the logs, and we can perform fastest searches when we need to retrieve certain information.

With the help of Confluence, we can generate reports of the test cycles that can be personalized according to the requirements of the user and they can have a historical of the executions they do so they can compare the results with the previous ones.

Our framework will execute the test cases automatically so this would save work and time for the testers and the developers of the application. The executions can be configurated to determine the number of cycles, test cases, android versions, devices and the data that will be saved in the logs.

1.5 Timeline

The features outlined above, combined with the students took approximately 6 months to complete. The primary focus is the development of a testing framework in python for mobile devices using deployment procedures, methodologies and best practices in the classroom with scrum.

Bringing a project as a topic of the class was something new, never implemented before at Panamerican University, it was hard to accurately estimate the exact time required to complete the project.

We had to teach the students the test automation processes and tools, the scrum process and get help from our principal consultant in order to develop the architecture and the code reviews. With these actions and after a few sprints the project was a success.

1.6 AMX Approach

Our project is a framework for automated mobile testing in Android devices developed with Python. The goal of the framework is to help the testers by reducing the time and the effort they put in automated testing.

The testers will create the test cases by using the methods of the framework and then they will determine the number of devices and the versions of the operative system they want to test. Finally, they will run the test cycle in the mobile devices that are connected to the farm and after this the framework will show a log of all the actions that were successful and the ones that failed.

This project has a Graphic User Interface for the user to see and interact with, so they don’t have to deal with the complications that can come when interacting with code.

The UP Framework is the core of this project. It has many classes with functions that are necessary to create the test cases, and with this we can do test automation in the application. We have a class for the mobile device, for the database, for android debug bridge (adb), for file management, for a logger that registers actions executed by the framework, for the basic actions you can do on a mobile app and for the mobile test farm.

The Test Suite is formed by the test cases that make specific tasks. They use the functionalities of the framework to develop a certain task. For example, they use the functionalities of starting a call, verifying it and end it for the task of making a call.

We used different Android devices like Motorola, Samsung and LG to test the application.

1.7 Structure

The following section contains an explanation of the different folders that compose the project, and the content of each one of them.

GUI Folder

An angular application that has the demo to execute the test cases of the framework.

Lib Folder

Adb Folder

Adb Controller Class has a function to run an adb command and other functions that will execute specific commands like install app, open app, close app, make phone call, download logcat, install app from PC, get the list of installed apps, toggle Bluetooth and take a screenshot.

Data Repository Folder

Data Device Class to create an object with the ID of the device and other properties like the brand, the operative system, the version of the operative system and others.

Data Repository Class that has functions to insert, update and to get all the records from the database that we have on Firebase.

File Manager Folder

File Manager Class that has functions to read and write files, to search a value on a file and to check if a directory exists on the device.

Logs Folder

Logger Class has functions that will write messages on the log file and a function to get the actual content of the logger. The type of messages are debug, info, success, warning and error.

My App Folder

My App Class to create an object with the package name of the app and it has a function to check it the app is opened.

My Device Folder

My Device Class to create an object with the ID of the device, the path of the logcat on the pc, an adb controller object, the path of the downloads on the PC and the path of the screenshots on the PC. It has functions to install an app, download an apk, install an apk from the PC, open an app, close an app, download log, open log, close log, press the back button, make phone call using the UI, clear apps, press the home button, check if an app is installed, toggle Bluetooth, toggle WIFI, end a call, pause a call, check if a call is answered in less than a minute, make a click, take a screenshot, copy files from device to PC, copy one file from device to PC and find a value on the log.

STF Device Folder

STF Device Class to create an object with the ID of the device, the URL of the mobile test farm and the token for the farm. It has functions to check if the farm is being used by a device, to add a device to the farm, to connect to a device from the farm, to disconnect a device from the farm, to get the connection address to the farm and to remove a device from the farm.

Benefits

Our framework can get testing done faster. This will help manual testers to concentrate on clever ways to find defects, instead of working on performance test inputs and verifying the output.

It also reduced the execution time by 70% in different devices and different android versions. With this result it increased test efficiency (productivity), test effectiveness and test coverage. It helps decrease the number of defects that escape to production and it helps improve test repeatability.

Our ROI projection is based on the execution of 91 cases (LTE Data) and on 10 devices.

AMX Framework Total Hours	480 hours (6 Sprints)
Effort Saved per Iteration	80
Total Manual Effort per device	8-13 hours (average)
Total Automated effort per device	5 hours (average)

References

Oracle - “When to Automate Your Testing (and When Not To)”

https://www.oracle.com/technetwork/cn/articles/when-to-automate-testing-1-130330.pdf